DPDK SUMMIT
Powering the Future of Networking Software

12–13 May 2026

PQC Integration in DPDK

OpenSSL PMD implementation and future offload roadmap

Gowrishankar Muthukrishnan
Akhil Goyal
Marvell Technology

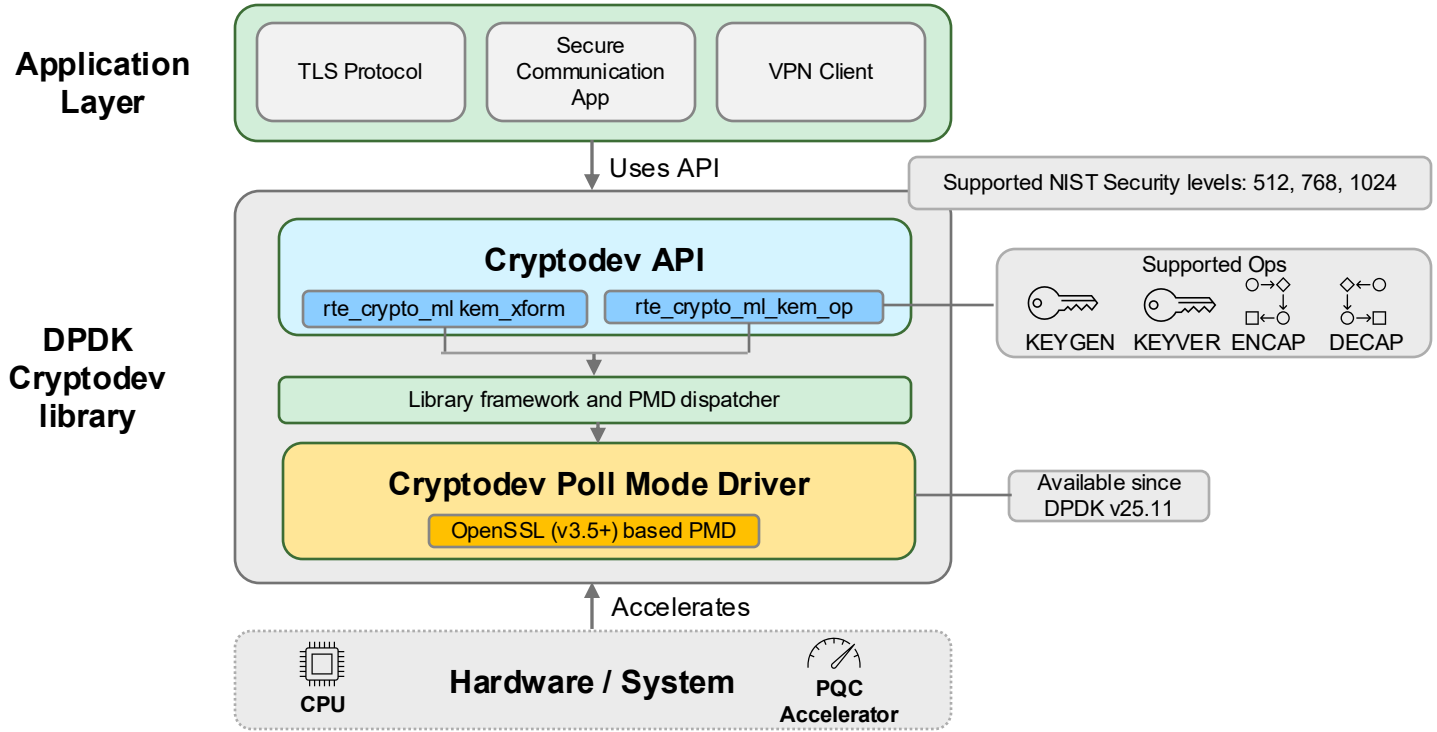
ML-KEM and ML-DSA

What is ML-KEM and ML-DSA ?
- Module Lattice based encryption and signature algorithms
- NIST standard (FIPS 203 and 204).
- Security based on MLWE (Module Learning With Error)
- Replaces classical crypto:
  ML-KEM replaces RSA, DH, ECDH
  ML-DSA replaces RSA, ECDSA

Why are they important in PQC ?
- SNDL (Store Now Decrypt Later) threat
- No compromise with the speed
- Interoperable (Browsers of different companies use same math)

What software enable these algorithms today ?
- Browsers, Messaging app, Core crypto libraries

ML-KEM API

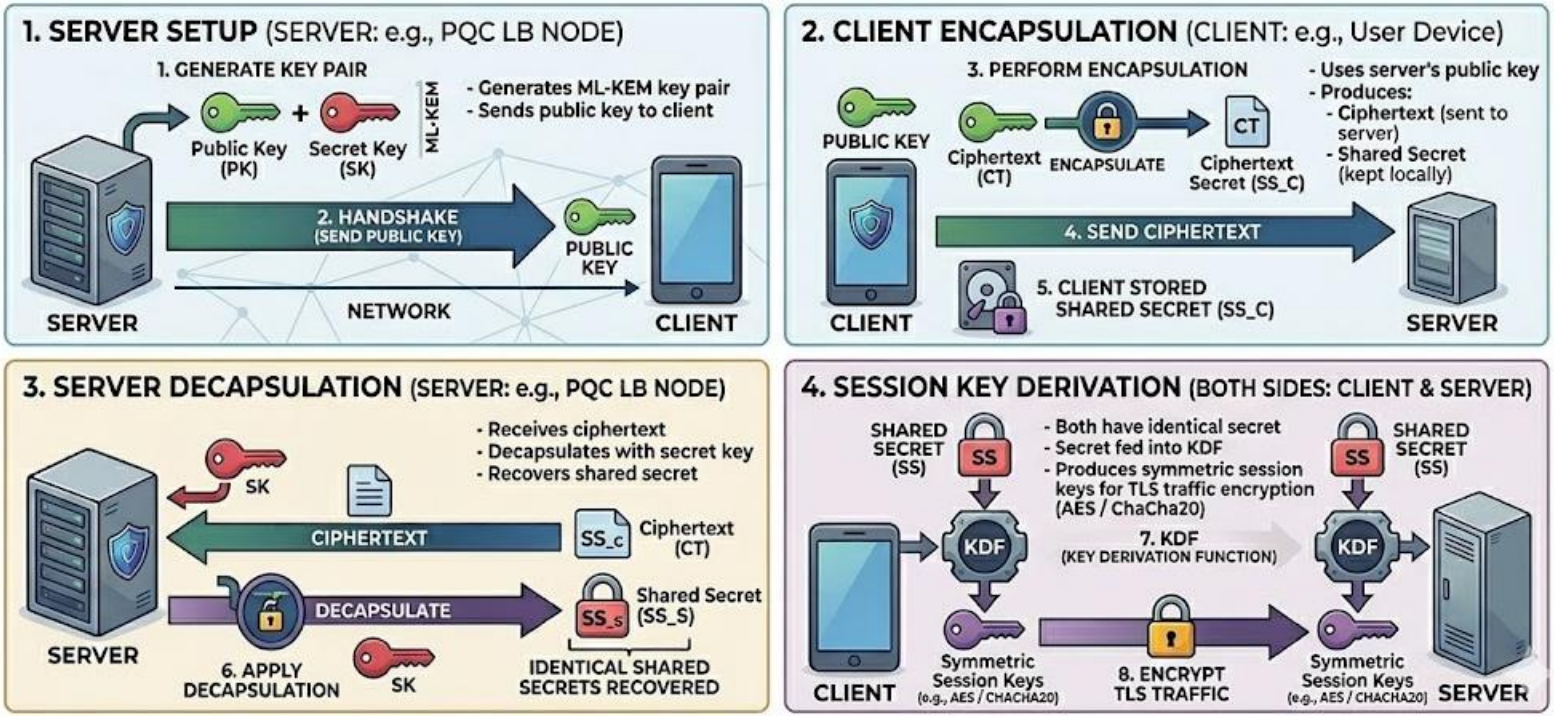

Application Layer
TLS Protocol
Secure Communication App
VPN Client
Uses API
Supported NIST Security levels: 512, 768, 1024
DPDK Cryptodev library
Cryptodev API
rte_crypto_ml kem_xform
rte_crypto_ml_kem_op
Supported Ops
KEYGEN
KEYVER
ENCAP
DECAP
Library framework and PMD dispatcher
Cryptodev Poll Mode Driver
OpenSSL (v3.5+) based PMD
Available since DPDK v25.11
Accelerates
CPU
Hardware / System
PQC Accelerator

ML-KEM op

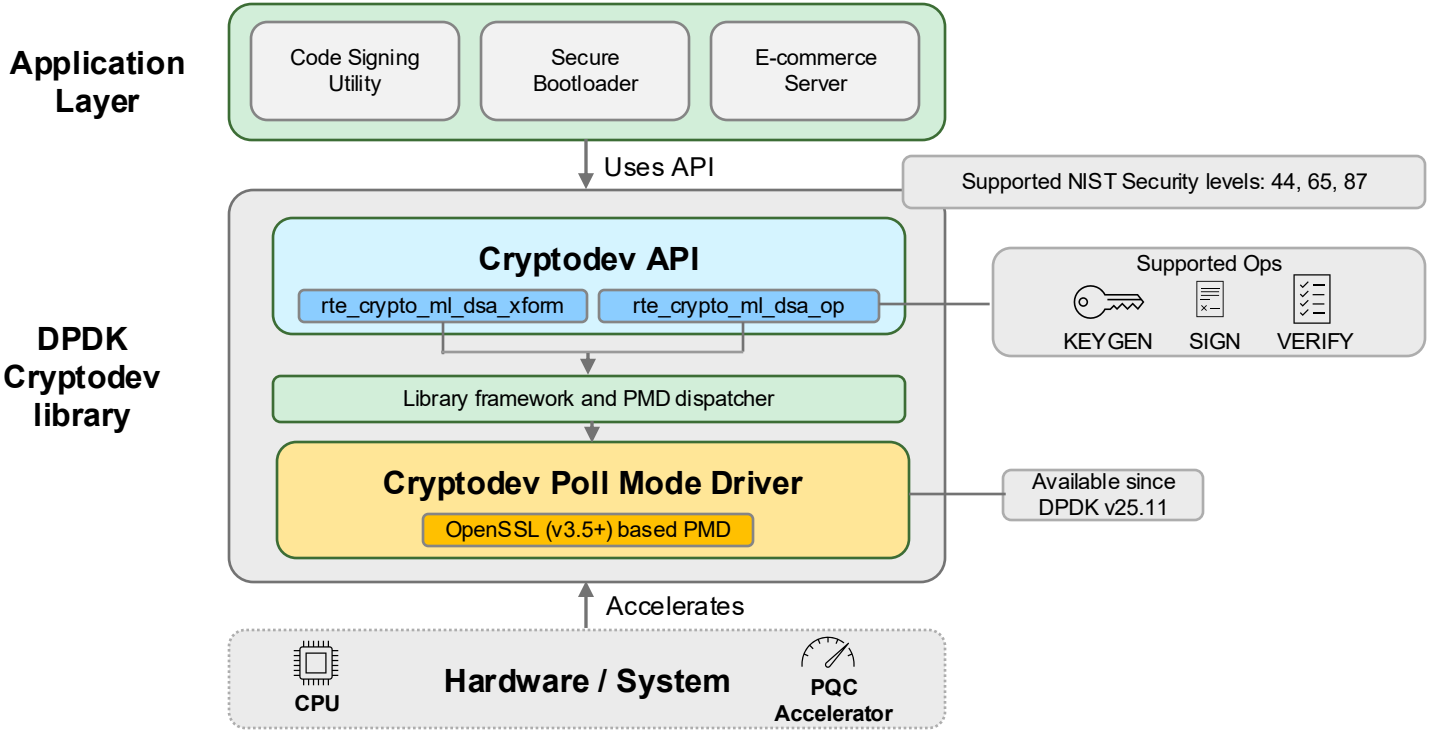
Ops	Key generation	Key verification	Encapsulation	Decapsulation
Input	Seed ‘d’ (32B) Seed ‘z’ (32B)	Op type Key	Message (32B) Encapsulation key	Cipher (32B) Decapsulation key
OpenSSL PMD	EVP_PKEY_fromdata_init EVP_PKEY_fromdata EVP_PKEY_todata	Not yet enabled	EVP_PKEY_encapsulate_init EVP_PKEY_encapsulate	EVP_PKEY_decapsulate_init EVP_PKEY_decapsulate
Output	Encap key Decap key	Success/Fail	Cipher Shared secret (32B)	Shared secret (32B)

Param set	Encapsulation key size	Decapsulation key size	Cipher size
ML-KEM-512	800	1632	768
ML-KEM-768	1184	2400	1088
ML-KEM-1568	1568	3168	1568

Case study: PQC enabled TLS Gateway



ML-DSA API

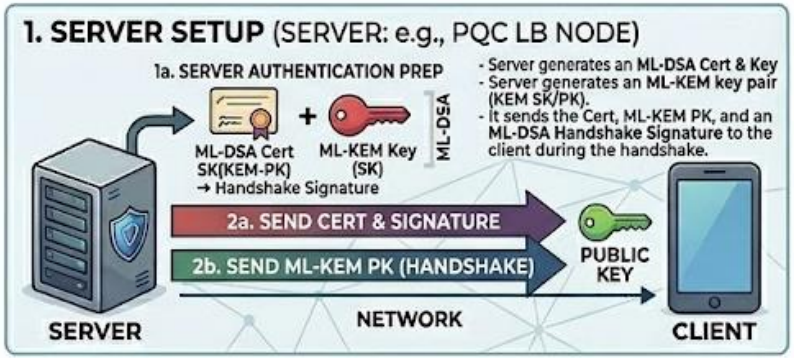

Application Layer
Code Signing Utility
Secure Bootloader
E-commerce Server
Uses API
Supported NIST Security levels: 44, 65, 87
DPDK Cryptodev library
Cryptodev API
rte_crypto_ml_dsa_xform
rte_crypto_ml_dsa_op
Supported Ops
KEYGEN
SIGN
VERIFY
Library framework and PMD dispatcher
Cryptodev Poll Mode Driver
OpenSSL (v3.5+) based PMD
Available since DPDK v25.11
Accelerates
CPU
Hardware / System
PQC Accelerator

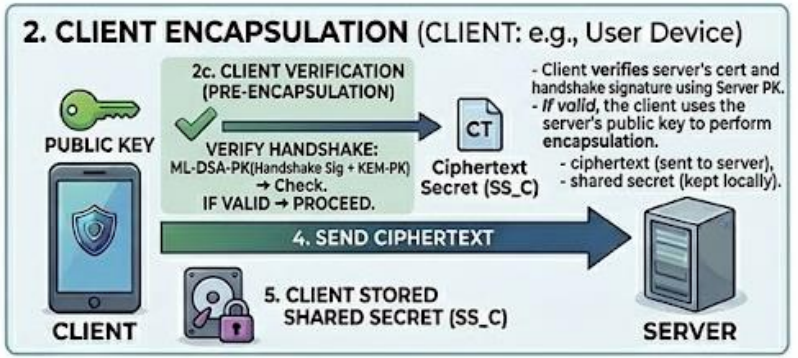
ML-DSA op

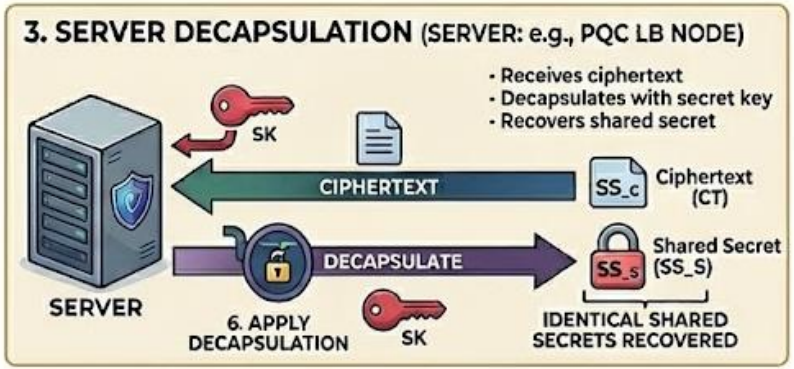
Ops	Key generation	Sign	Verify
Input	Seed (32B)	Message Mu Private key Per signature seed Context key Hash algorithm	Message Mu Public key Signature Context key Hash algorithm
OpenSSL PMD	EVP_PKEY_fromdata_init EVP_PKEY_fromdata EVP_PKEY_todata	EVP_PKEY_sign_message_init EVP_PKEY_CTX_set_signature_md EVP_PKEY_sign	EVP_PKEY_verify_message_init EVP_PKEY_verfiy
Output	Public key Private key	Signature	Success/Failure

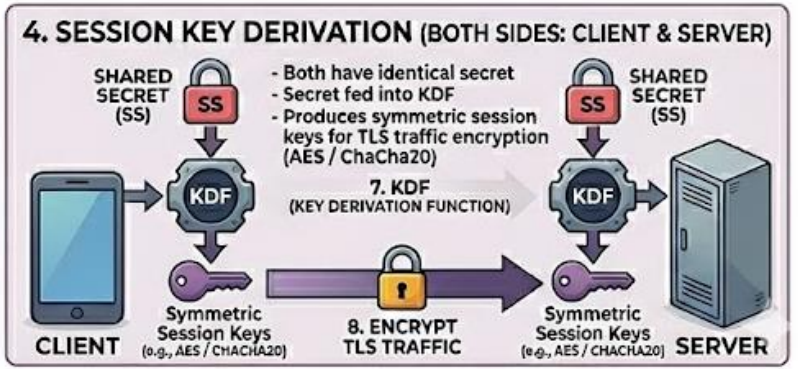
Param set	Public key size	Private key size	Signature size
ML-DSA-44	1312	2560	2420
ML-DSA-65	1952	4032	3309
ML-DSA-87	2592	4896	4627

Case study: PQC authenticated TLS Load Balancer

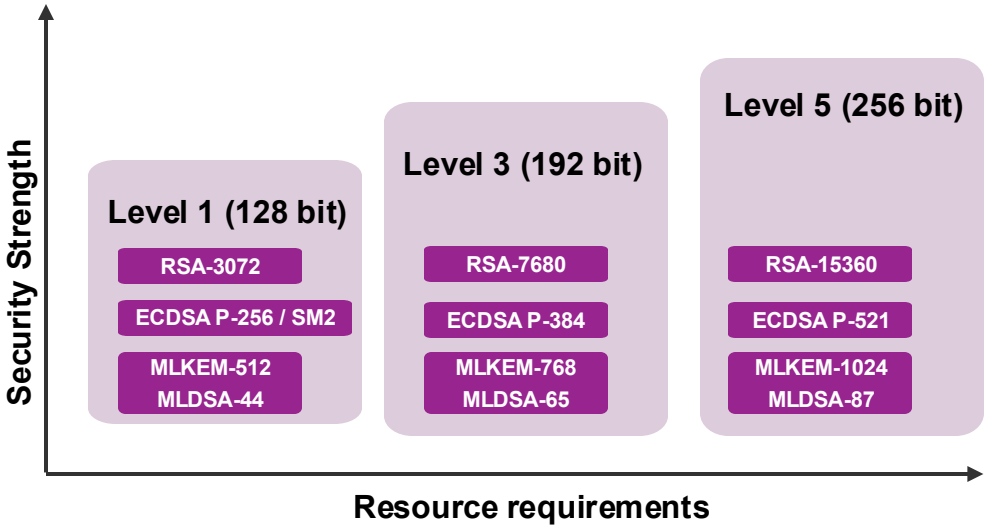








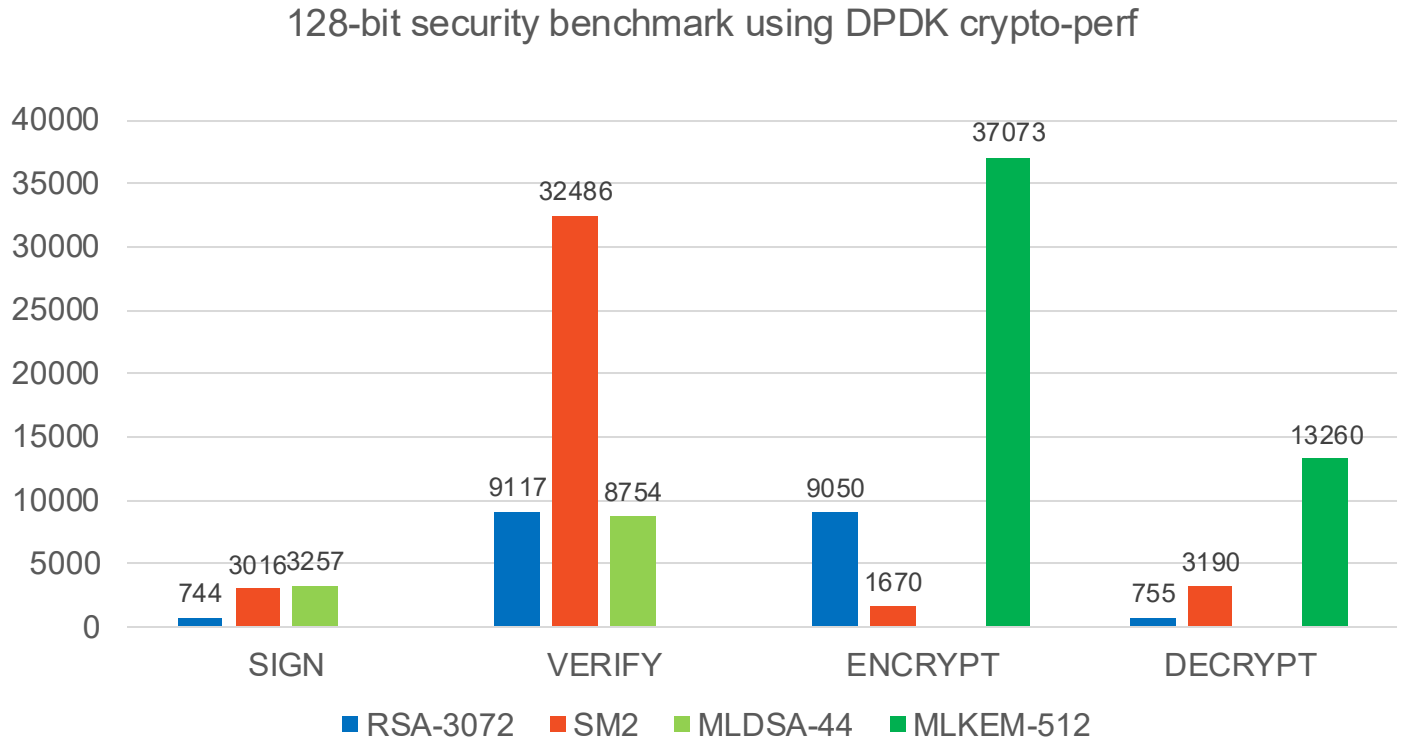
PQC adoption challenges



Challenges:

1. **Memory allocation**
   Larger key and signature sizes in handshake.
2. **Cache efficiency**
   Larger keys thrash cache.
3. **Session table**
   Shared secret needs to be stored per session.
4. **Network throughput**
   Packet fragmentation on handshakes.

Software crypto performance



Based on OpenSSL v3.6 based DPDK PMD in x86_64, 3GHz x8 cpu, 16GB Memory

Future roadmap for DPDK + PQC

Hardware acceleration
- Significant performance improvements are anticipated in the modern hardware platforms.

Cryptodev to support Fn-DSA, SLH-DSA
VirtIO crypto to support PQC algorithms
Hybrid PQC
- Support dual key encapsulation
- Use cases RFC 7512/PKCS 11 , RFC 9370/IKEv2)

Benchmark Quantum Tax
- Throughput vs. Latency
- MTU challenge (larger key size ~ 2 – 3 KB)

OpenSSL PMD improvement
- Batch processing of KEM and DSA workloads

Questions.


DPDK
SUMMIT
Powering the Future of Networking Software